CSU 670

September 7, 2005

Software Development
CSU 670
Organization

- **INSTRUCTOR** Professor Karl J. Lieberherr, 308A West Village H, (617) 373-2077, electronic mail: lieber AT ccs.neu.edu.

  To communicate with the teaching assistant, please send mail to csu670-grader@ccs.neu.edu which also reaches the instructor.

- **TEXTS**

  Required book:

  1. AUTHOR = "Andrew Hunt and David Thomas", TITLE = "The Pragmatic Programmer", PUBLISHER = "Addison-Wesley", YEAR = "2000", ISBN = "0-201-61622-X".
  2. Class Packet from NU Bookstore.

  Recommended books (depends on your interests):

  1. "Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns" by Karl J. Lieberherr, PWS Publishing Company, ISBN: 0-534-94602-X. Selected chapters are in the class package. Also available on the Web from the Demeter home page. As an alternative to buying the book, you could print selected chapters on your home printer or download into your laptop.
  2. (you may use your favorite Java book) Ken Arnold and James Gosling, The Java Programming Language, Addison Wesley.
  3. Design patterns by Erich Gamma et al., Addison Wesley.
  4. How to Design Programs by Felleisen, Findler, Flatt, Krishnamurthi, MIT press.
  5. Java Language Specification by Gosling et al., Addison Wesley.

- **PREREQUISITE** (COM 1204 or CSU370) AND (COM 1350 or CSU390).

- **COURSE REQUIREMENTS** The grade will be based on an open-book midterm (twenty per cent) and an open-book final examination (thirty per cent), and the homework solutions (twenty percent) which are primarily programming exercises and the project (thirty percent).

- **GOALS AND THEMES**

  A message from a reader of the course textbook summarizes what the course tries to achieve:

```
Date: Mon, 20 May 1996 15:49:38 -0400 (EDT)
To: demeter@ccs.neu.edu
Subject: Thanks for Demeter

The subject really gives it away. Thanks for Demeter.
A few weeks I stumbled across the book in a local bookstore and
it somehow drew my attention. Now after having studied it
I find myself employing
the philosophy in the projects I am doing.
...
I think this book for the first time provided really convincing material
in favor of OO programming.
I especially like the structure shyness of the approach.
...
Although I follow the methodology I am
more than eager to use the tools.

Mark van der Voort
May 20, 1996
Eindhoven, Netherlands
```

  The following message indicates that what you learn is not only useful for Java/C++ programming. It is a programming language independent.

```
Subject: Java version of Demeter

Thank you for all the work you have been doing. It has had a profound
impact on my understanding of objects, and has helped me educate and train
many Smalltalk students.

Demeter seems to be that combination of the
best science with the finest aesthetics in the service of a practical
```

```
art which is good engineering. It deserves recognition.

Are there any plans to port the tools to Java? This would be a very popular
platform, I am sure.

Steve Parker
From sparker@well.com Fri Mar  8 01:23:37 1996
```

Knowledge of the Demeter Method will be useful in the following ways after this course:

- In your current or next job you will be asked to program in Smalltalk, another popular object-oriented language. Since you have learned object-oriented design and programming in a programming-language independent notation, you can easily transfer your knowledge. You will have to express your programs in a different syntax.
- In your current or next job you will be asked to design a class library for modeling the business of loan processing at a local bank. You impress your manager by providing a highly graphical design which she or he can understand and you also impress by your English-like object descriptions for the important objects in the loan processing business.
- In your current or next job you are given a Java program written by someone who was not an experienced object-oriented designer and who has left the company. You are asked to rewrite the program so that it will be easier to maintain. You impress your manager by writing the program with a 75% size reduction and you impress your manager by your speed of adjusting your program to new requirements.
- In your current or next job you are asked to translate from a stylized English command language to a French version of the command language. The next day you come back with a program which solves task, and you ask your manager for a raise.
- In course X in CCS you are given a programming assignment in your programming language of choice. You solve the task in Java using DemeterJ and you turn in the most elegant solution.

The goal of the course is to significantly improve the software development skills by teaching useful abstractions for proper separation of crosscutting concerns.

- **FEEDBACK FROM FORMER STUDENTS AND USERS OF ADAPTIVE PROGRAMMING**

```
From file

ftp://ftp.ccs.neu.edu/pub/people/lieber/Demeter-interest
```

THIS IS A COLLECTION OF MESSAGES ABOUT
THE USE AND POTENTIAL OF THE ADAPTIVE
SOFTWARE TECHNOLOGY AND THE DEMETER TOOLS/C++
==

The high-level approach of Adaptive Software is very interesting. I believe the biggest benefit in using Adaptive Software is the next level of abstraction up from Object-Oriented programming.

Doug Sprague
IBM Microelectronics
Dec. 1995
===================================

From the ACM Computing Reviews, October 1995, page 528.

About the Communications of the ACM paper:
''Adaptive Object-Oriented Programming using Graph-Based Customization'',
by Karl Lieberherr, Ignacio Silva-Lepe and Cun Xiao,
{\em Communications of the ACM}, May 1994, pages 94-101.

Quote from review:
"If you have been writing object-oriented programs and are interested in what comes next, this paper is certainly for you."

Wayne Summers, Las Vegas, NM
Oct. 1995
===

I might add that propagation patterns are a way to specify flexible (C++) database programs which adapt to a broad class of schemas or, another way to look at it, allow the application programmer to specify code that does not need to know in detail the database schema.

Given that in practical C/S application for MIS, DSS, TP and in general mission critical applications, the underlying database can have a very complex schema, this has the potential, in my opinion, of greatly simplifying writing C++ mission critical C/S applications.

Professor Ugo Gagliardi
Harvard University
May 30, 1995

======================================

From an anonymous reviewer of a funded NSF proposal on
"Engineering adaptive software"
(award for $295000 to Northeastern University)

I have followed the work of Professor Lieberherr for a number of years, and have been consistently impressed by the approach and the results. He and his group have had many good ideas in the area of object-oriented programming and development methodology, and have published them. The Law of Demeter is widely recognized as an important stylistic guide. They also have an excellent record of going beyond the ideas and their publication to implementation and use in an educational environment and beyond. The Demeter System supports at least two object-oriented programming languages, extending its benefits to programmers in those languages without requiring them to learn a new one. It has been used in many classes at Northeastern University, and has been requested by many other organizations. Several tutorials have been offered on the Demeter Method, and it has been included in a survey of object-oriented analysis and design methods undertaken by the Object Management Group.

This proposal deals with a critically important area of research. The object-oriented paradigm is widely believed to provide major benefits, and is becoming increasingly widely used. Actually achieving the desired benefits of reuse, extensibility and ease of maintenance, however, is no easy matter. Simply using object-oriented style is not enough. Many researchers therefore study object-oriented methodologies to find ways to improve those characteristics. Many of the approaches, such as the popular and valuable "frameworks" approach, allow the good designer to produce a unit of software that is highly reusable in certain planned ways. Such a unit of software will not easily be reusable in unplanned ways, however. Only a few approaches to object-oriented software engineering concentrate on supporting unplanned extensions. These tend to require extensions to the object-oriented paradigm, so they are not currently "mainstream", but it is my belief that these approaches will be of paramount importance as object-oriented software engineering matures, and will be instrumental in achieving its potential. Prof. Lieberherr's "adaptive software" approach is one of these.

Anonymous NSF reviewer

=================================

It has long been a goal in the software development community
to link the design documents and the implementation code more directly.
The Demeter System is a powerful tool that helps designers
accomplish this goal. Because so much of the conversion from design to
implementation is made automatic by Demeter, when errors are detected, the
designer is encouraged to modify the design rather than tinker with the
low level C++ code. In the context of a software design course,
this methodology encourages students to think and design on an
abstract level rather than on the level of a
particular programming language such as C++.
In our view, this is precisely the kind of thinking
that computer science educators have been trying to instill in
students for more than 20 years.
...
Given the fact that students can learn only a single methodology
in detail [in an eleven week term], we believe that Demeter is a particularly
suitable one for teaching purposes. The reason is that Demeter is based
on theoretical studies of the optimal way to design loosely
coupled classes for maximum effectiveness and flexibility.
These theoretical principles can be used by students in whatever design
environment they happen to work in. Comments by students who have completed the
Software Design and Development course indicate that this is in
fact the case.

Richard Rasala
Professor and Associate Dean of Undergraduate Education
College of Computer Science, Northeastern University, Boston
May 1, 1995

=================================

My approach was to define the Library System class dictionary. Once the
class dictionary was designed, the next step was to define the operations
that would be performed on that class dictionary.
Lastly I designed the propagation patterns to implement those Library functions.

Although this class was my first introduction to Object Oriented Design
and to C++, I found that the Demeter System and adaptive S/W to be extremely
flexible and a very powerful technique. This flexibility was important
especially in this project when I had to make changes (of which there were
many).

I never had any problems with the basic Demeter tools: gen-imake, gen-make,

make, run, make clobber, and make clean.

Peter Monteagudo
Hardware Design and Development Engineer
Raytheon Company
Missile Systems Division Labs in Tewksbury, MA
Dec. 11, 1994
=================================

Overall, we learned that adaptive software allowed us to implement a fairly complex software system in a fast, flexible, and efficient manner. It reduced the amount of time devoted to actual coding and debugging, and allowed us to redesign parts of the package with very little effort.

Rafael David
Sarah Sullivan
Dec. 11, 1994

=====================================

The adaptive
software work by Karl Lieberherr and his group is
also very useful here because it's about telling you
something about the way the internal data structures
are arranged, but not too much about how they are
arranged.
...
and again, adaptive software plays in here
because it is about being looser about the carving of the world.

Gregor Kiczales, Xerox PARC,
in his invited presentation at OOPSLA '94
(see http://www.xerox.com)
==================================

From an anonymous course feedback form:

Question:
Do you agree or disagree with the view that class dictionaries and propagation patterns are a design notation which you can manually translate into the programming language which your employer has decided to use?
Give an explanation.

Answer:
Agree. Learning the adaptive software notations ...
has done wonders for my object-oriented programming without
the Demeter Tools.

Undergraduate student in the Northeastern
Software Design and Development class
==

The project I am working on ... is developing an object-oriented power
system model in C++ for ABB Network Control in Vasteras.
When the implementation of the model was almost done, I read your article on
Demeter C++ in the Feb. 94 issue of the C++ report. It struck me that with
a method and tools as you describe in the article, I would have been able
to reduce the amount of time I spent on implementing and debugging my
power system model drastically. It seems to me that the Demeter C++
system addresses exactly those aspects of C++ development that make
implementing and comprehending a non-trivial C++
application tricky and time consuming.

Is a commercial product ready today?

Martin Falkevall
Magistratsv 55 K324
S-226 44 Lund
SWEDEN

==
I believe that the Demeter approach is a significant advance
in the art of software development ... because it allows the construction
of object-oriented systems, but reduces the dependence of
programming on the schema of the object dictionary.

Terry Glagowski
Washington State University
June 7, 1994
==
I've found Demeter to be very useful in regards to compiler design. For this
homework, I developed a grammar using Demeter and 'sem-check -i' for an
object-oriented language that is LL(1) compliant. The translation to EBNF
form was then trivial. sem-check's analysis and violation reporting made the
design of the grammar much faster.

The grammar was then ready for input into the compiler-compiler LLgen:

...

The scanner definition, designed for flex, is very similar to that of Demeter's generic parser. In fact, I used many of the patterns in lex-DEM.l for my scanner.

...

I know Demeter is useful for many more things besides compiler design, but that is my main interest. Traversal and transportation specifications, along with wrappers, make coding such things as intermediate-code generation and optimization analysis much easier than tools such as yacc or bison.

Andrew Purtell
College of Computer Science, Northeastern University
June 9, 1994

In response to the CACM paper, May 1994, pages 94-101.

... Adaptive software leaves ...
the issues of partitioning the world into objects
and navigation between objects flexible.

I completely agree with your perception that this is a serious issue.
...
It is also one that holds a lot of interest for the group I work in ...

...
We discussed your CACM paper today at our weekly paper reading group, and came up with a couple of ways of looking at your system other than the "adaptive" description:

> "It's a tool that lets you do OO programming directly in terms of class dictionary graphs, bringing programming one level closer to OOA/OOD."

> "It's a tool supporting more powerful forms of object relative reference in OO programming than just slot access."

John Lamping
Xerox Parc

around May 19, 1994
========================
"I am very enthusiastic about the Demeter System, and feel it represents a major advance in software engineering (a field where "hard" advances are rare).
I think it will also transform the way a number of areas of computer science are taught."
...
I've been using the software fairly heavily and have a few comments on it. ...
(I've been using it to teach a class and my students are using Demeter to design and build a text-adventure game). It is a remarkable system with many potential applications beyond the scope of software engineering (for instance Compiler Design can be done very effectively using the Demeter System.)

"This is an extraordinary powerful software tool that will undoubtedly become a standard fixture in Computer Science curricula (and in several areas, including Compiler Design, Software Engineering, Artifical Intelligence)."
...
"Extremely valuable --- I predict it will sweep the industry"

Justin R. Smith
Drexel University
May 19, 1994
========================
From a discussion on the patterns news group.

The Visitor class reminds me of the Demeter work by Karl Lieberherr and his associates at Northeastern University. They're taking an approach that involves abstracting the "meat" of the behavior of a complex construct from the "skeleton" methods that deal with managing and navigating the construct. Their work also addresses keeping the abstraction valid across changes to the implementation of the construct. To do this, they use a graph notation to represent the class structure, and a "propagation pattern" language that tells, in a general way, how to distribute the meaningful methods and navigation methods across the structure. A propagation pattern may be applicable to many class structures, thus allowing changing the latter without needing to modify the former. A code generator does the work of creating the C++ code. A good introduction to this is in an article called "A Report on Demeter C++" in the February '94 issue of C++ Report, by Ignacio Silva-Lepe, Walter Huersch, and Greg Sullivan.
If you can't get to this periodical,
you might try emailing them at {nacho,huersch,gregs}@ccs.neu.edu.

```
Don Dwiggins                    "Things should be made as simple as possible,
Mark V Systems, Inc.               but no simpler"
dwig@markv.com                       -- Albert Einstein
April 7, 1994
============================
```

```
I really think you are on to something with Demeter.
I think propagation patterns are very powerful.
...
I think that the best part of Demeter is the "adaptive" part.  Being
able to modify the interface of a class or change the class hierarchy
without major problems is a huge benefit of propagation patterns.  C++
and OO languages in general do not address this problem.

Joseph N. Coco, Software Engineer
TASC, 55 Walkers Brook Drive, Reading, MA  01867

Dec. 20, 1993
```